Simple UI, Intents & Interactivity

CSC2007 & ICT2105 Mobile Application Development Spring 2021

Overview

This lab provides guidelines to create a simple interactive app, and gain familiarity with the Android Activity class, Layouts, simple View classes and Intents.

Outcomes

Upon completion of the session, you should be able to:

- Create a simple Android interactive interface using Layouts and View classes
- Create and send Intents to invoke another Activity or another application
- Create simple layouts and use UI elements

Build a Simple User Interface

The following are exercises to create and run a new Android Application and project as described in the previous practical session and described in the following links:

https://developer.android.com/courses/basic-android-kotlin-training/overview OR

https://developer.android.com/training/basics/firstapp/creating-project
https://developer.android.com/training/basics/firstapp/running-app

Following the instructions and steps below, create your first simple user interface:
https://developer.android.com/training/basics/firstapp/building-ui

Add more code to build an Intent and start another Activity:
https://developer.android.com/training/basics/firstapp/starting-activity

Question: How do you commit the completed code to your own repository in Github?

Learn about Views, Activities & Intents

The following are structured exercises to learn about Activities, ScrollViews and Intents. Follow the instructions and steps in the links below:

https://codelabs.developers.google.com/codelabs/kotlin-android-training-app-anatomy/index.html#0

https://codelabs.developers.google.com/codelabs/kotlin-android-training-linear-layout/index.html#0

https://codelabs.developers.google.com/codelabs/kotlin-android-training-interactivity/index.html#0

https://codelabs.developers.google.com/codelabs/android-training-activity-with-implicit-intent/index.html (This lab is in Java, but the concepts are the same)

Question: What's the difference between an explicit and implicit Intent?

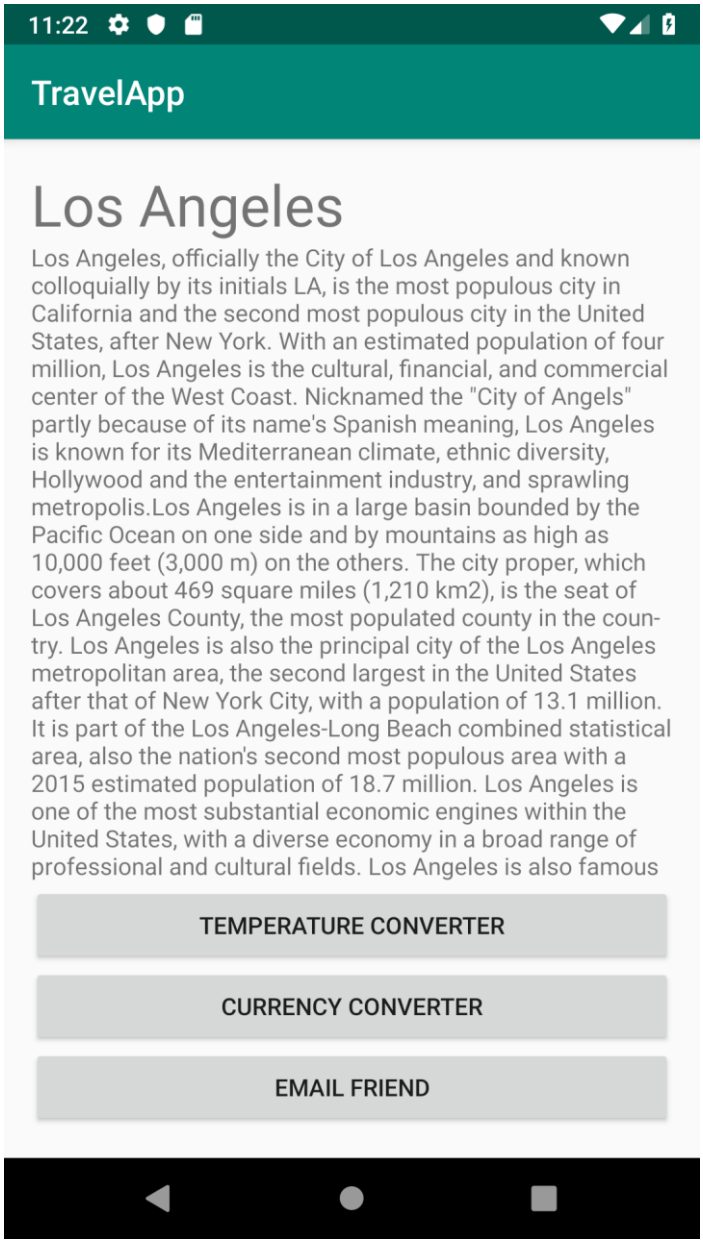
Creating a Travel Application

This section provides exercises to create a simple travel application. It consists of a user interface to display destination information, do temperature conversion, currency conversion and send email. The goal is to create a user interface to display text and implement currency and temperature conversion as given in the screenshots.

Fork the repo **ict2105-lab02-2021** or **csc2007-lab02-2021** and inspect the code within the project.

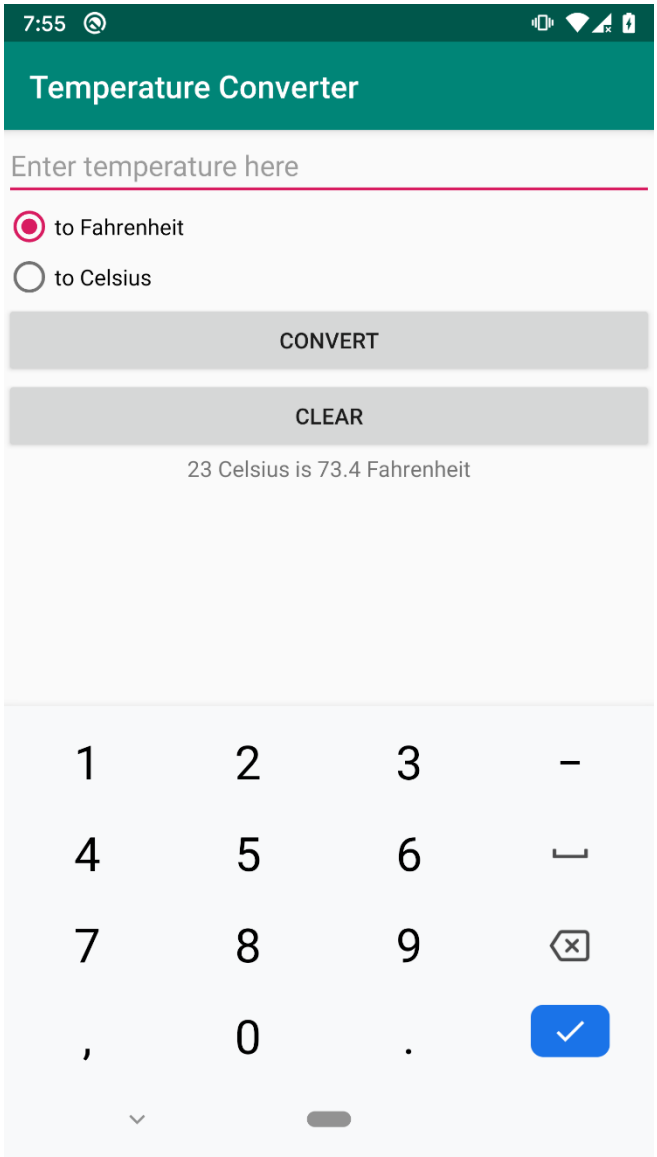
In the AndroidManifest.xml file for the app, add the correct elements so that this application can use the CurrencyConverterActivity, EmailActivity and TempConverterActivity.

On the MainActivity, implement a header and scrollview, and create 3 buttons that will launch the other three activities, CurrencyConverterActivity, EmailActivity and TempConverterActivity.



Design the layouts of each of the screens similar to the following screenshots. Experiment with ConstraintLayout, RelativeLayout and LinearLayout to achieve the look.

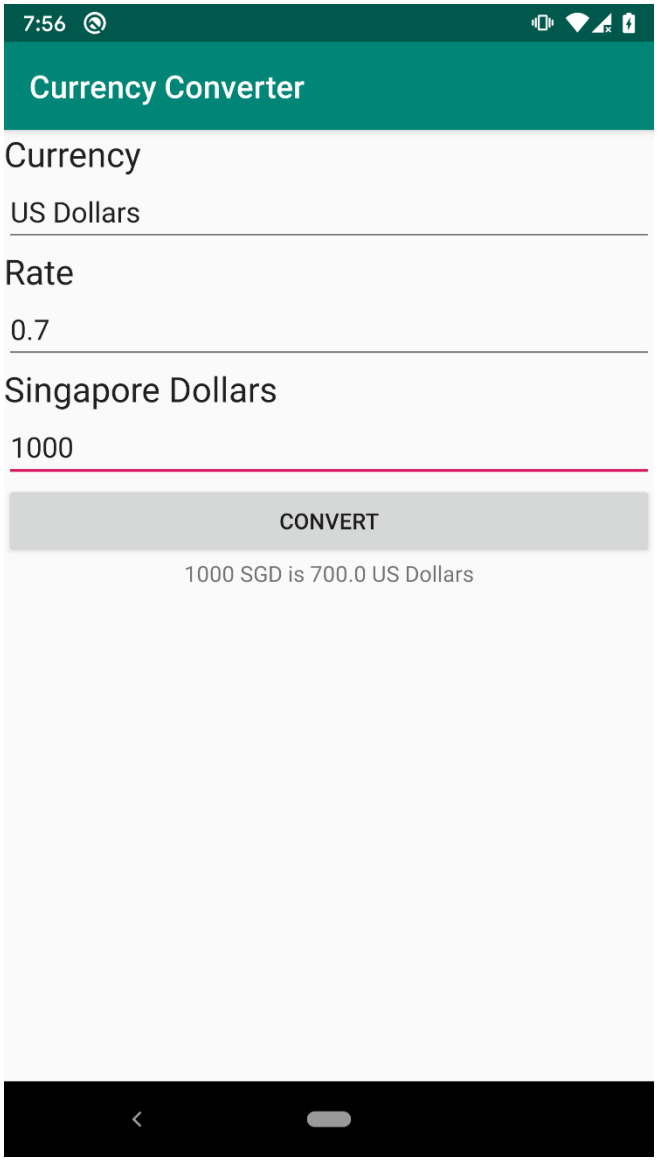
IMPORTANT: Ensure there are no spelling errors on the button text. Ensure that the header TextView id is `headerTextView`, and the ScrollView id is `destinationScrollView` within the XML .



In TempConverterActivity, implement the interface, and the logic to do the temperature conversion within the method **`convertCelsiusToFahrenheit()`.** What's the formula to convert from Celsius to Fahrenheit and vice-versa? The text field (EditText) id should be named **`editTextTemp`.** Display the result within the text below the button. Pressing the

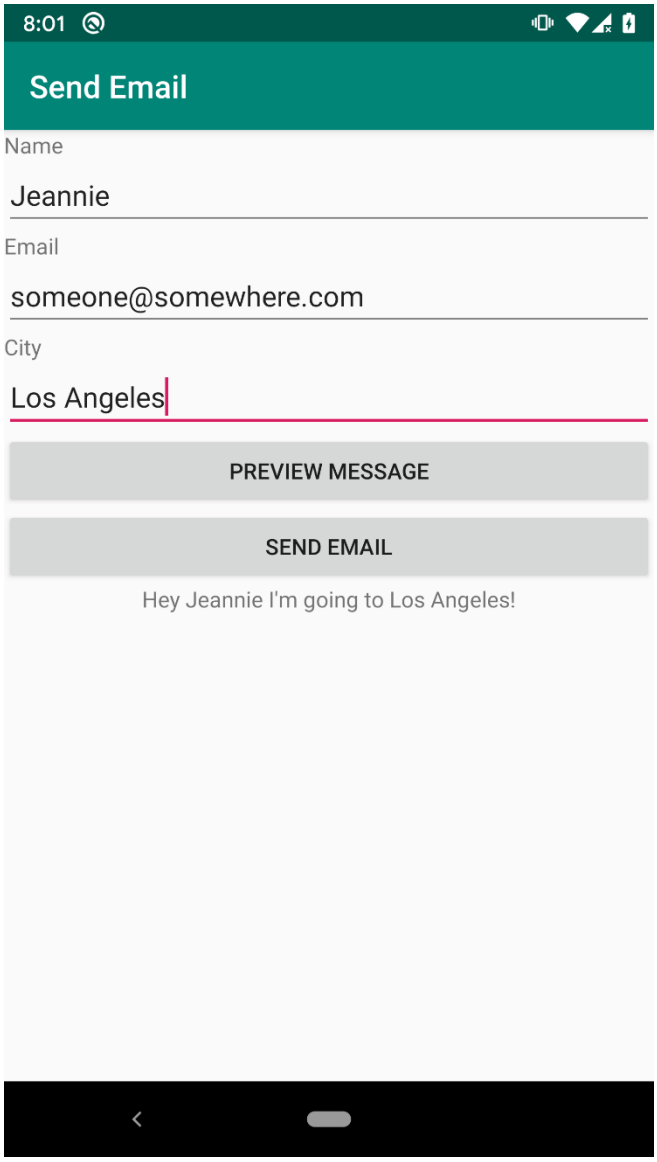
"Clear" button clears the text field. Implement code to retain the text after orientation change.

IMPORTANT: Ensure that the EditText id is editTextTemp within the XML .



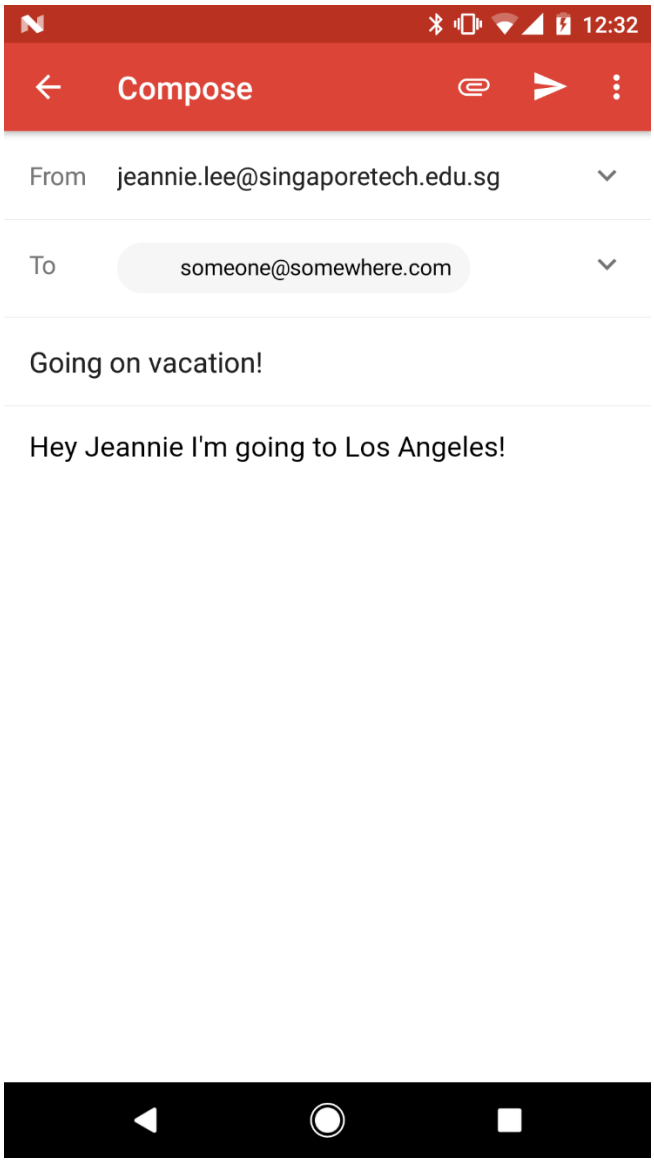
For the CurrencyConverterActivity, implement the interface and the logic to take the input from the text fields and compute the correct currency conversion within the method `calculateRate()` and display it within the text below the button. The text field (EditText) ids should be named **editTextCurrency, editTextRate** and **editTextSingDollar** respectively**.** Implement code to retain the text on all the fields after orientation change.

IMPORTANT: Ensure that the EditText ids are `editTextCurrency`, `editTextRate` and `editTextSingDollar` within the XML .



For the EmailActivity, take the input city or country name, and the friend's name and use the method to create the sample message string. The text field (EditText) ids should be named **editTextName, editTextEmail** and **editTextCity** respectively**.** Then take the valid email address, and create an Intent to launch the default email application (normally Gmail). Place the email address in the To: field, place the message within the body of

the email, and set the subject as, "Going on vacation!" **IMPORTANT: Ensure that the EditText ids are `editTextName`, `editTextEmail` and `editTextCity` within the XML .**



Screenshot of the default email application (normally Gmail) that is launched after the "Send Email" button is pressed. The email address and subject "Going on vacation!" are inserted, and the message is placed within the body of the email.

Lab Exercise 2

Due Date: Mon Feb 1, 2021 2359hrs (CSC2007)
Thu Feb 4, 2021 2359hrs (ICT2105)

Completion of all the tasks as listed above, which are summarized below:

1. Fork the repo **ict2105-lab02-2021** or **csc2007-lab02-2021**
2. Implement the missing tags within AndroidManifest.xml to launch the correct activities.
3. Design the layouts of each of the four screens similar to the given screenshots. Experiment with ConstraintLayout, RelativeLayout and LinearLayout to achieve the correct look.
4. MainActivity: ScrollView and Button interface. The header TextView id should be named **headerTextView**, and the ScrollView id is **destinationScrollView** within the XML.
5. Implement the logic and interface on each of the screens:
    a. CurrencyConverterActivity: Currency conversion logic and interface
    The text field (EditText) ids should be named **editTextCurrency, editTextRate** and **editTextSingDollar.**
    b. TempConverterActivity: Temperature conversion logic and interface
    The text field (EditText) id should be named **editTextTemp.**
    c. EmailActivity: Launch the email application with the given input strings
    The text field (EditText) ids should be named **editTextName, editTextEmail** and **editTextCity.**
6. Commit and push all changes to your forked repository:
    ict2105-lab02-2021 or **csc2007-lab02-2021**

IMPORTANT: Please ensure the button and text spellings are correct, and the UI element names are correct.

END OF DOCUMENT